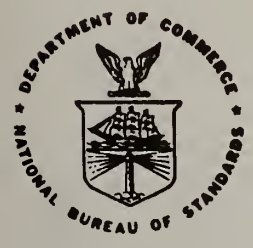
NBSIR 83-2763

Artificial Traffic Generation of ISO Transport Class IV Protocol Data Units on an IEEE 802.3 10 Megabit CSMA/CD Local Area Network

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Institute for Computer Sciences and Technology
Systems and Network Architecture Division
Washington, DC 20234

July 1983



U.S. DEPARTMENT OF COMMERCE
NATIONAL BUREAU OF STANDARDS

NBSIR 83-2763

ARTIFICIAL TRAFFIC GENERATION OF ISO TRANSPORT CLASS IV PROTOCOL DATA UNITS ON AN IEEE 802.3 10 MEGABIT CSMA/CD LOCAL AREA NETWORK

Timothy Gardner

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Institute for Computer Sciences and Technology
Systems and Network Architecture Division
Washington, DC 20234

July 1983

U.S. DEPARTMENT OF COMMERCE, Malcolm Baldrige, *Secretary*
NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Director*

Table of Contents

1. Abstract

The National Bureau of Standards' Institute for Computer Sciences and Technology (ICST) is testing the International Organization for Standardization's (ISO) Transport Class IV protocol on an Institute of Electrical and Electronics Engineers (IEEE) 802.3 Local Area Network. Part of the test facility includes an artificial traffic generator that produces ISO Transport Class IV protocol data units encapsulated in IEEE 802.2 Type 1 Class 1 Logical Link frames. The traffic generator submits the frames to the network for transmission. This document describes the architecture and usage of the traffic generator for multi-host and multi- connection traffic.

Keywords: computer networks; CSMA/CD; local area networks; traffic generation; standards; transport protocols

2. Introduction

The National Bureau of Standards' Institute for Computer Sciences and Technology (ICST) prepared specifications and produced an implementation of the International Organization for Standardization's (ISO) Transport Class IV Protocol. At the request of a number of companies, ICST held a series of workshops [1,2] for local network vendors interested in implementing and testing these specifications [3]. In conjunction with this effort, a multi- vendor demonstration of ISO Transport operating on a local area network conforming to IEEE 802.3 Local Area Networking specifications [4] is expected in the 1984 time frame.

ICST is developing several protocol test tools including tools that test the correctness of the Transport protocol implementation [5,6,7] and performance measurement tools designed to assist in the testing, tuning and debugging of vendor implementations. This document discusses the traffic generator used by the performance measurement tool now under construction.

This new tool (see figure 1) consists of a data acquisition system, a real-time measurement facility and an analysis and display software subsystem. These components interconnect and attach to an IEEE 802.3 CSMA/CD local area network. The data acquisition system captures Transport, Network, and Link protocol data units from the network and passes protocol header information to the real-time measurement subsystem. The real- time measurement facility tracks the transport protocol's finite state machine and produces measures related to aggregate, link, and transport traffic, host to host traffic, and connection traffic. These measures are passed on to the analysis and display subsystem where they are analyzed and displayed, providing vendors and users with applicable tables, histograms, pie charts, and other graphical aids for determining the performance and for tuning the transport protocol.

This document provides a detailed description of the traffic generation system (TGS) used by ICST to test the above mentioned protocol test tool. Section 3 gives a general overview followed by two sections that descibe each of the individual components of the system. Appendix 1 explains how to use TGS to generate traffic. It is intended as a user's guide for those with access to the actual system.

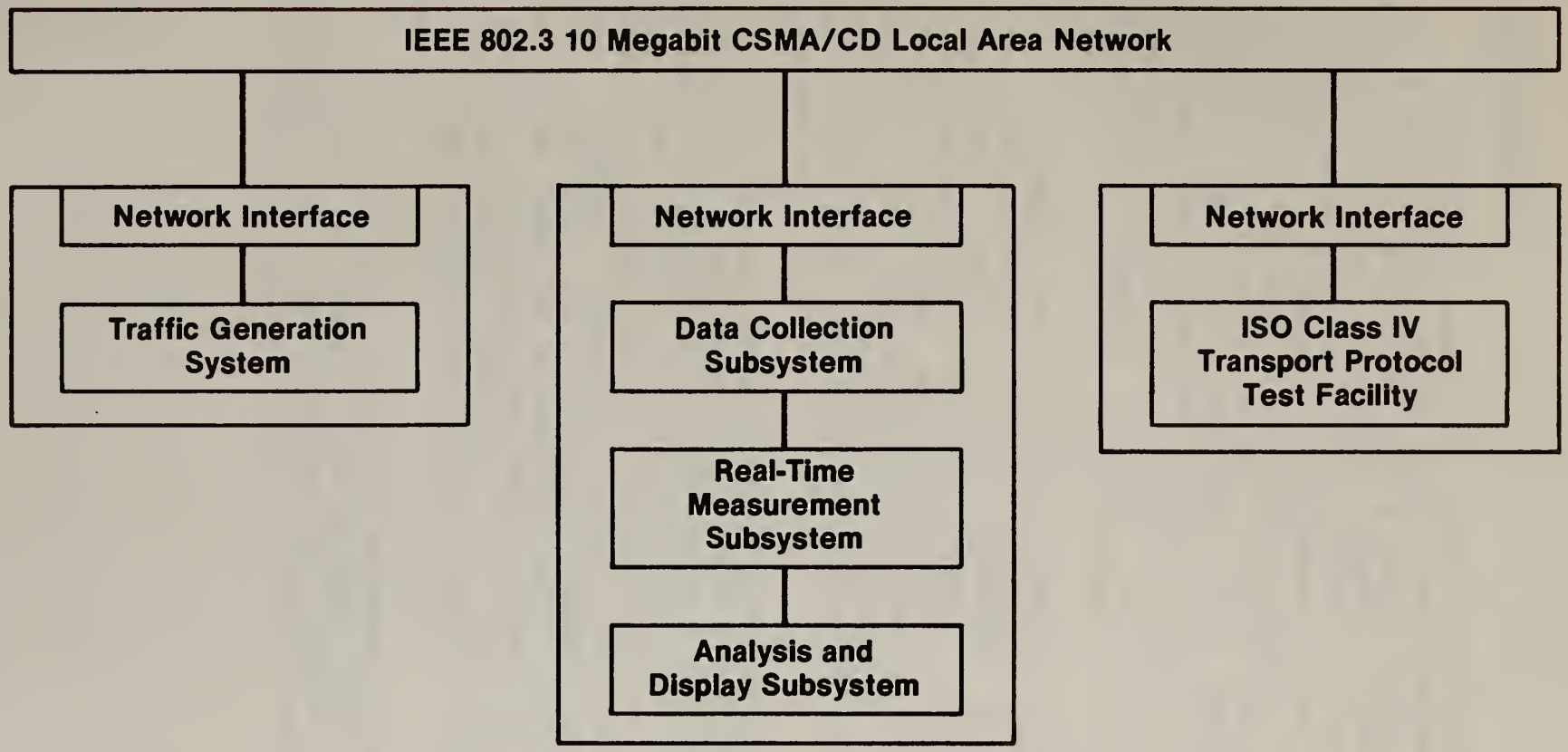


Figure 1. ICST Protocol Test Facility and Network Performance Measurement Setup

3. TGS Overview

The traffic generation system described by this report is a system designed to create, edit, store, interpret, and transmit IEEE 802.2 [8] link frames containing ISO transport class IV protocol data units (TPDUs) as data. The generator is capable of producing protocol data units containing any destination or source link address and any set of transport destination and source reference numbers. In addition, the generator optionally creates "blocked" TPDUs, i.e. multiple TPDUs encapsulated in one IEEE 802.2 Logical Link frame. TGS generates flexible traffic useful in the testing and tuning of programs using traffic from an 802.3 local area network by allowing a user to control the flow and the content of the traffic being created.

The TPDUs used by TGS for creating artificial traffic are created by an exception generator which is part of the correctness testing tools mentioned above [5]. This exception generator creates "correct" transport protocol and stores a log of the TPDUs in files called eglogs. TGS modifies the destination and source reference numbers in each TPDU and generates Link source and destination addresses thus creating multi-connection and multi- host network traffic.

TGS consists of a group of programs and a group of data files where the programs store and retrieve data. The two principle programs in the system are the Command File Editor and the Traffic Simulator/Generator. The Command File Editor is responsible for creating the set of commands that the Traffic Simulator uses to create the "traffic environment", i.e. the set of connections, the connection entities, and the contents of each connection. The Traffic Generator then transmits packets from this environment to the local area network. The remaining programs in the system are utility and support programs that make the system more versatile.

The following gives a brief description of each program and group of data files in TGS. Refer to figure 2 while reading the descriptions to better understand the architecture of the Traffic Generation System.

TGS consists of the following programs and data files:

Programs:

Filter - removes non-transport material from exception generator log files (eglogs). Input is binary eglogs. Output is binary scenario files.

Command - menu driven program that assists user in developing commands to run the traffic simulator. Input is user responses to prompts, the ascii scenario directory file - "scenarios", and existing binary command files. Output is binary command files.

Simgen - menu driven program that assists user in creating traffic simulations and in generating network traffic from these simulations. Input is user responses to prompts, binary command files, files in the scenarios file block, and binary simulation files. Output is binary simulation files and network traffic.

Interp - creates an ascii interpretation of simulation files for the purpose of debugging TGS and the analysis programs that use TGS traffic. Input is binary simulation files. Output is an ascii interpretation.

Convert - converts ascii interpretations of scenario files created by Tlook back into binary scenario files. Input is ascii Tlook output. Output is binary scenario files.

Tlook - creates an ascii interpretation of binary scenario files and eglogs. This permits a user to modify scenario files and convert the ascii version back to binary using Convert. Input is binary eglogs and binary scenario files. Output is ascii interpretations.

Data files:

eglogs - Eglogs are created by the protocol lab exception generator. An eglog contains a group of TPDUs that make up a transport scenario.

comfiles - binary command files that contain traffic simulation commands. These files are created by the command file editor and are used to run traffic simulations.

scenarios - this block contains binary scenario files and an ascii directory file. Scenario files are eglogs that have have been filtered to remove non-transport material. The directory file contains the name, reference number, and description of every scenario file. This file is maintained by the user and updated whenever a scenario file is added to or deleted from the block. The files in this block are used by Simgen for the creation of traffic simulations. The directory file called "scenarios" is used by the command file editor for creating and editing command files.

simfiles - binary simulation files containing output from the traffic simulator. These files are also used by the traffic generator for network traffic generation.

asciiscens - ascii interpretations of binary scenario files. These files are created by Tlook. Because these files are ascii, a user may edit them and then convert them back to binary scenario files using Convert.

4. Program Descriptions

4.1 Filter

Filter removes non-transport material from eglogs created by the exception generator. Non-transport material consists of information the exception generator outputs prior to the connect request TPDU of a connection. The material is stored in the same format as a TPDU, but the value of the event code in the header line (see 5.1) is different. Filter removes all material not containing an event code of hexadecimal 6 or hexadecimal 16.

The input to Filter is an eglog. The output is a scenario file.

To run Filter: filter [eglog] [scenario file]

4.2 Command

Command is the command file editor used to create and edit command files to drive the traffic simulator (Simgen).

Input to Command is user responses to prompts, command files, and the scenarios directory file. Output from Command is command files, user prompts, and user information.

To run Command: command

The screen displayed when the program is run is the start menu. The user is prompted for one of the following three commands.

c - create command file
e - edit existing command file
q - quit program

The following describes the above commands.

c - Command prompts the user for a file name. If the file entered already exists, Command notifies the user that it can not overwrite existing files and the user is prompted for one of the three commands. If the file does not exist the editor menu is displayed.

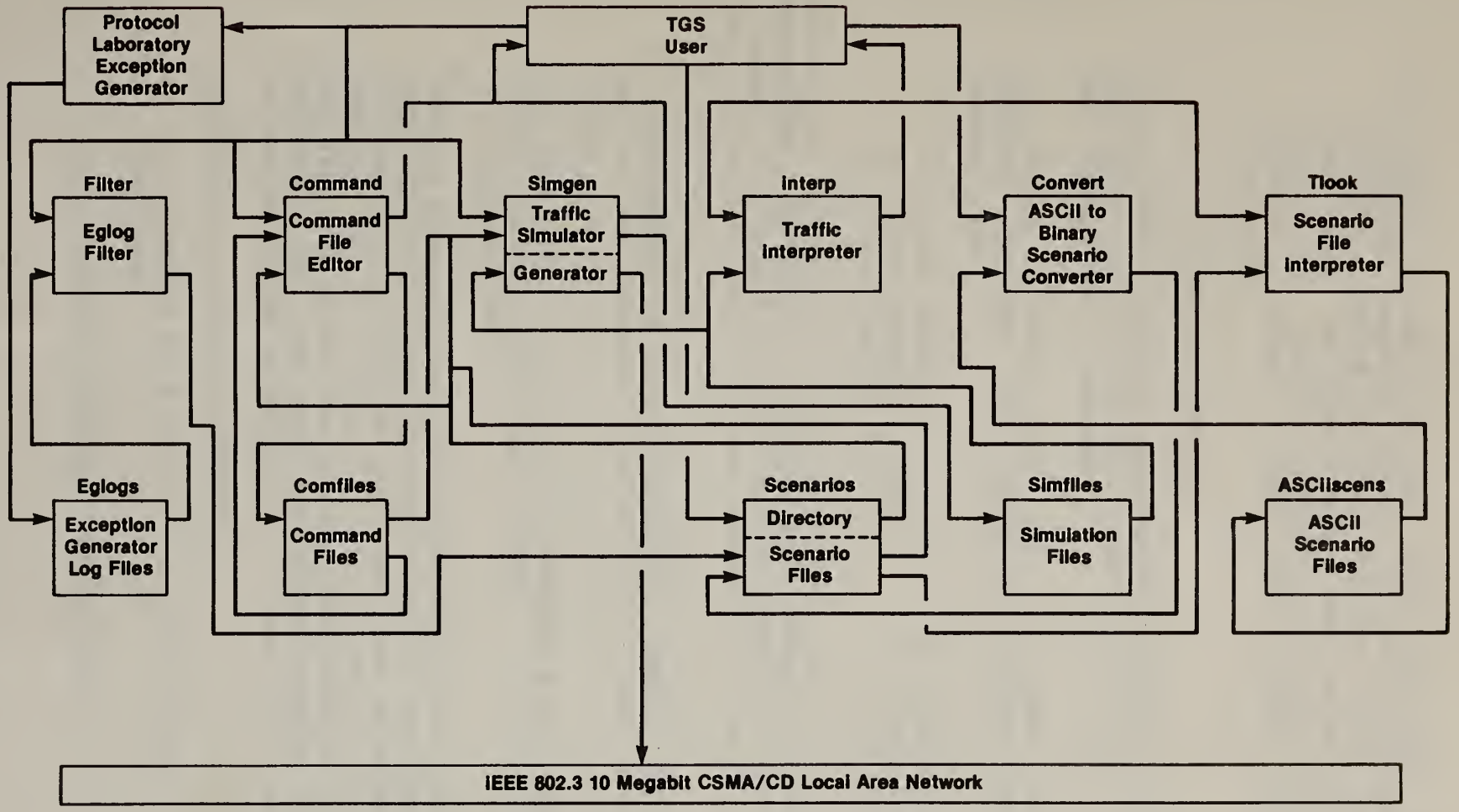


Figure 2. Overview of the Traffic Generation System

e - Command prompts the user for the name of an existing command file. If the file cannot be opened, Command notifies the user that it can not open the file and the user is prompted for one of the three commands. If the file is opened without error, the editor menu is displayed.

q - the program terminates.

If any other response is entered, Command notifies the user that an illegal response has been entered and prompts for a command.

After a legal file name has been entered, Command displays the following editor menu and header lines:

e - exit editor	d - delete a connection
s - list scenario files	i - insert a connection
c - list command file	r - refresh display

connection number	scenario file #	connection host #	between and host #	connection repeats	packet blocking

If the editor is in edit existing command file mode, the chosen command file is listed under the header lines.

The last line of the screen contains the prompt line: "Enter connection number to edit or CR for next connection".

At this time the user may enter any one of the menu commands When prompted for any other response, only commands "e", "s", or "c" may be entered. If command "d", "i", or "r" is entered when prompted for anything other than a conection number, Command notifies the user that the chosen command can only be executed when prompted for a connection number.

Description of menu commands:

e - the command file is written and the start menu is displayed.

s - the scenario file menu is displayed.

The scenario file menu contains the following commands:

m - return to main menu
l - list scenario files
c - continue listing

Description of scenario file commands:

m - the user is returned to the editor.

l - a descriptive listing of all scenario files is listed.

c - the next page of scenario files is listed if another page exists. If no additional pages exist, the user is notified that the listing is complete.

l - the list command file menu is displayed.

The list command file menu is identical to the list scenario files menu and functions in the same manner except that instead of listing scenarios, the command file being edited is listed. This menu allows a user to view prevous pages of the command file if they exist.

d - Command prompts the user for a connection number to delete. If an existing connection number is entered, the connection is deleted; otherwise the user is notified that the entered connection cannot be deleted.

i - Command prompts the user for the number of the connection the new connection is to be inserted above. If an existing connection number is entered, the user is prompted for the scenario file number of the new connection; otherwise, the user is notified that a connection cannot be inserted.

r - the display is refreshed showing only the current status of each connection.

If just a carriage return is entered, Command prints the number of the next connection to be formed and prompts the user for a scenario file number.

If the number of an existing connection is entered, the information about the connection is displayed and the user is prompted for the scenario file number.

If an illegal command or the number of a nonexistant connection is entered, the user is notified of an illegal response.

After entering a carriage return (CR), an existing connection number, or inserting a new connection, the user is prompted for a scenario file number. A list of available scenario file numbers is seen by entering the "s" command. A user is not limited to the scenario numbers listed by the "s" command though. If the connection is being edited, a CR may be entered to continue on to the next prompt without changing the displayed value. This is true for all prompts after the connection number prompt.

The next two prompts are for the number of the host initiating the connection and for the host receiving the connection. Host numbers are only reference numbers that are mapped into network addresses by the traffic simulator. A host number is limited to integers between 1 and 6 inclusive. An illegal response message is displayed if a number greater than 6 or less than 1 is entered.

Connection repeats indicate to the traffic simulator how many times to repeat the connection upon termination. If an "*" is entered the connection will repeat forever.

Packet blocking indicates whether or not TPDUs from the connection may be blocked together or with TPDUs from other connections. The user should enter a "y", or a "n" as a response to this prompt.

After a legal response is entered for the packet blocking prompt, the connection number prompt is once again displayed.

4.3 Simgen

Simgen is the traffic simulator and generator used for creating traffic simulations and generating network traffic.

Input to Simgen is user responses, command files, scenario files, the scenario block directory file, and simulation files. Output is user prompts, user information, simulation files, and network traffic.

To run Simgen: simgen

Simgen contains only the following menu that is displayed when the program is run:

```
c - create a traffic simulation
t - run traffic generator from existing
    simulation file
x - change network addresses of hosts
q - quit program
```

Description of menu commands:

c - the user is prompted for a command file. If the command file cannot be opened, the user is notified and prompted for a command. If the command file is opened successfully, the user is prompted for a simulation file. If a CR is entered, the simulator output is not sent to a simulation file. This option is useful when using command files that contain repeat forever connections. If the simulation file entered already exists, the user is notified that Simgen cannot overwrite existing simulation files and prompted for another simulation file.

After a legal response to the simulation file prompt, the user is asked if the output should be sent to the network. If a "y" is entered Simgen tries to open the network interface. If an error occurs, the simulation is aborted and the user is prompted for a command. Otherwise, the simulation is started.

If at anytime during the simulation an error is found in the command file or elsewhere, the user is prompted for instructions. A user may chose to abort the simulation or continue the simulation without the connection causing the error. The user is notified upon the successful completion of the simulation.

t - the user is prompted for a simulation file. If the simulation file cannot be opened, the user is notified and prompted for a command. After a legal simulation file is entered, Simgen tries to open the network interface. If an error occurs, the user is notified and prompted for a command. Otherwise, the traffic generation is started and the user is notified upon completion.

x - a list of the default or current network addresses is displayed. If the user enters a CR, the address is not changed and the prompt moves to the next address. If an illegal address is entered the user is notified that an illegal address has been entered and once again prompted for an address. After a legal response for the last address has been entered, the user is prompted for one of the menu commands.

q - Simgen is terminated.

4.4 Interp

Interp creates ascii interpretations of binary simulation files. These interpretations are useful for debugging TGS and for verification of the network traffic generated to test other programs using TGS traffic.

Input to Interp is simulation files. Output is an ascii interpretation outputted to standard output.

To run Interp: interp [simulation file]

If an error occurs trying to open the simulation file, Interp notifies the user and terminates.

4.6 Convert

Convert converts asciiscens files back into binary scenario files.

Input to Convert is asciiscens files. Output is scenario files.

To run Convert: convert [asciiscens file]
[scenario file]

If an existing scenario file is given as the scenario file argument, Convert notifies the user that it cannot overwrite existing scenario files and terminates. If an error occurs trying to open the asciiscens file, Convert notifies the user and terminates.

4.5 Tlook

Tlook creates ascii interpretations of binary scenario files. These files may then be edited to control the content of the traffic being generated. Tlook may also be used to create ascii interpretations of eglogs.

Input to Tlook is scenario files or eglogs. Output is an asciiscens file.

To run Tlook: tlook [scenario file or eglog]
[asciiscens file]

If the asciiscens file argument is not given, output is to the terminal. If an existing asciiscens file is given as the asciiscens file argument Tlook notifies the user that it cannot overwrite existing asciiscens files and terminates. If an error occurs trying to open the scenario file, Tlook notifies the user and terminates.

5. File Descriptions

5.1 Eglogs

Eglogs are created by the Network Protocol Group exception generator and are used as input by Filter and Tlook.

Binary eglog format:

byte	field		field description
+0	tsize(0)	tsize(1)	total size of record
+2	acode(0)	acode(1)	exception generator code
+4	secs(0)	secs(1)	timetag seconds
+6	secs(2)	secs(3)	
+8	msecs(0)	msecs(1)	timetag milliseconds
+10	head(0)	head(1)	eg header 12 bytes
.	.	.	header bytes
+20	head(10)	ecode(11)	event code - part of header
+22	TPDU(0)	TPDU(1)	TPDU tsize - 22 bytes
.			TPDUs
+tsize	TPDU	fill	fill exists if tsize is odd

5.2 Comfiles

Comfiles are created by Command and are used as input to Command and Simgen.

Binary comfile format:

byte	field		field description
+0	sfnum(0)	sfnum(1)	scenario file number
+2	host1(0)	host1(1)	initiating host number
+4	host2(0)	host2(1)	receiving host number
+6	reps(0)	reps(1)	connection repeats
+8	blkg(0)	blkg(1)	TPDU blocking

5.3 Scenarios

This group of data files consists of an ascii directory file containing a description of each of the scenario files and the binary scenario files.

The ascii directory file is created by the TGS user and is used as input by Command and Simgen.

Ascii directory file format:

column	to column	field	field description
1	6	blank	blank
7	16	name	scenario file name
17	36	number	scenario file number
37	48	numTPDUs	number of TPDUs in file
49	EOLN	descrp	file description

Scenarios are created by Filter and are used by Singen and Tlook.

Binary scenarios format:

same as format of binary eglogs

5.4 Simfiles

Simfiles are created by Simgen and are used by Simgen and Interp.

Binary simfile format:

byte	field		field description
+0	len(0)	len(1)	total record length
+2	daddr(0)	daddr(1)	network destination address
+4	daddr(2)	daddr(3)	
+6	daddr(4)	daddr(5)	
+8	saddr(0)	saddr(1)	network source address
+10	saddr(2)	saddr(3)	
+12	saddr(4)	saddr(5)	
+14	lgth(0)	lgth(1)	802 length
+16	ctrl(0)	net(1)	control and network
+18	TPDU(0)	TPDU(1)	TPDUs
.			
+len	TPDU	filler	filler if len is odd

5.5 Asciiscens

Asciiscens are created by Tlook and are used as input by Convert.

Ascii asciiscens format:

	column to column		field	field description
line 1	1	10	timetag	hours:min:sec.tenths
	11	11	blank	blank
	12	19	(size =	ascii characters
	20	21 or 22	size	TPDU size including header
	22 or 23	30 or 31	, code =	ascii characters
	31 or 32	31 or 32	code	exception generator code
	32 or 33	32 or 33	)	ascii character
line 2	1	3	header	hex header byte
	4	6	header	hex header byte
	.	.		
	28	30	header	hex header byte
line 3	1	3	header	hex header byte
	4	7	header	hex header byte
	8	10	TPDU	hex TPDU byte
	11	14	TPDU	hex TPDU byte
.	.	.		

Lines continue in same format with 10 pairs of TPDU bytes per line until the end of the TPDU. TPDU pairs are separated by 2 blanks; TPDUs in a pair are separated by 1 blank.

Records are separated by a blank line.

Appendix 1

Artificial Traffic Generation

The first step in using TGS to generate traffic is obtaining an eglog. Use the following steps to obtain an eglog:

1. Login to the Protocol Laboratory computer.
2. Select scenario to be used to create eglog. A listing of eg scenarios is in /u2/night/phase2/testdoc/g.appl. Each scenario has 2 files associated with it. These files reside in /u2/night/phase2 /newrt and /u2/night/phase2/newtc. Copy the rt version and the tc version to the user directory.
4. In rt version of the file, change the first 2 numbers of the first non-commented line from 1 2 to 2 1.
5. Enter the following commands:

```
tstart -el
next 0 /u2/gardner/tgs
next 0 tc_scenario name
next 1 rt_scenario name
```

6. An eglog now exists in a file called eg.log. Move this file into the tgs/eglogs directory on the system containing TGS using available FTP programs.

A much more detailed explaination is found in [5].

The next step is to filter the eglog to create a scenario file. This step is explained in 4.1.

Since a new scenario file has been added, the scenarios directory file must be updated. This is done by editing tgs/scenarios/scenarios and adding the information about the new scenario file following the format in 5.3.

Now run tgs/programs/command to create a command file. A detailed explaination of how to run Command is found in 4.2.

Run tgs/programs/simgen as explained in 4.3. This is the final step in creating network traffic.

The traffic created by Simgen may be interpreted if Simgen was run in the create simulation file mode. To interpret the simulation file, run tgs/programs/interp according to the instructions in 4.4.

To change a scenario file, run tgs/utility/tlook according to 4.5. The asciiscens file created will reside in tgs/asciiscens and may now be edited. When editing is complete, convert the asciiscens file back into a scenario file using tgs/utility/convert. An explaination of how to run this program is found in 4.6. Once again, the scenarios directory file must be update since a scenario file was changed or added.

A copy of this documentation resides in the tgs/docs directory and the source code of all programs reside in tgs/source.

References

[1] Proceedings of the First LAN-Transport Workshop. Nat. Bur. Stand. (U.S.) NBSIR 83-2673; 1983 February.

[2] Proceedings of the Second LAN-Transport Workshop. Nat. Bur. Stand. (U.S.) NBSIR 83-2717; 1983 March.

[3] Proposed Federal Information Processing Standard, Specifications of a Transport Protocol for Computer Communications, Volumes 1-6. Nat. Bur. Stand. ICST/HLNP 83-1 - ICST/HLNP 83-6; 1983 February.

[4] Draft IEEE Standard 802.3 CSMA/CD Access Method and Physical Layer Specifications. Institute of Electricial and Electronic Engineers. 1982 December.

[5] Nightingale, J. Stephen. Users Guide to the Testing System for Implementations of the ICST Transport Protocol. Nat. Bur. Stand. ICST/SNA 2; 1983 July.

[6] Nightingale, J. Stephen. A Test Suite for Implementations of the ICST Transport Protocol. Nat. Bur. Stand. ICST/SNA 3; 1983 July.

[7] Nightingale, J. Stephen. Specifications of a Remote Scenario Interpreter for Implementations of the ICST Transport Protocol. Nat. Bur. Stand. ICST/SNA 3; 1983 July.

[8] Draft IEEE Standard 802.2 Logical Link Control. Institute of Electrical and Electronic Engineers. 1982 November.

NBS-114A (REV. 2-8C)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions)	1. PUBLICATION OR REPORT NO. NBSIR 83-2763	2. Performing Organ. Report No.	3. Publication Date August 1983

4. TITLE AND SUBTITLE

Artificial Traffic Generation of ISO Transport Class IV Protocol Data Units on an IEEE 802.3 10 Megabit CSMA/CD Local Area Network.

5. AUTHOR(S)

Timothy J. Gardner

6. PERFORMING ORGANIZATION (If joint or other than NBS, see instructions)

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, D.C. 20234

7. Contract/Grant No.

8. Type of Report & Period Covered

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (Street, City, State, ZIP)

10. SUPPLEMENTARY NOTES

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)

The National Bureau of Standards' Institute for Computer Sciences and Technology (ICST) is testing the International Organization for Standardization's (ISO) Transport Class IV protocol on an Institute of Electrical and Electronics Engineers (IEEE) 802.3 Local Area Network. Part of the test facility includes an artificial traffic generator that produces ISO Transport Class IV protocol data units encapsulated in IEEE 802.2 Type 1 Class 1 Logical Link frames. The traffic generator submits the frames to the network for transmission. This document describes the architecture and usage of the traffic generator for multi-host and multi-connection traffic.

12. KEY WORDS (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)

computer networks; CSMA/CE; local area networks; standards; traffic generation; transport protocols

13. AVAILABILITY

☒ Unlimited

☐ For Official Distribution. Do Not Release to NTIS

☐ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.

☒ Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES

21

15. Price

$7.00